PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
G06K 15/00, G06F 3/12, G06T 11/00

A1

(11) International Publication Number: WO 99/19830

(43) International Publication Date: 22 April 1999 (22.04.99)

(21) International Application Number: PCT/US98/21587

(22) International Filing Date: 13 October 1998 (13.10.98)

(30) Priority Data:

60/062,348	15 October 1997 (15.10.97)	US
Not furnished	7 October 1998 (07.10.98)	US

(71) Applicant: PIXO ARTS CORPORATION [US/US]; Suite 110, 2570 West El Camino Real, Mountain View, CA 94040 (US).

(72) Inventors: LIU, Clive, K.; 25426 Adobe Lane, Los Altos Hills, CA 94022 (US). VASILEV, Dmitriy, N.; 14362 Springer Avenue, Saratoga, CA 95070 (US). SINGH, Rarendra, P.; P-250 Jalvayo Vihqr, Gurgaon, Haryana 1222001 (IN).

(74) Agent: SWERNOFSKY, Steven, A.; The Law Offices of Steven A. Swernofsky, P.O. Box 390013, Mountain View, CA 94039-0013 (US).

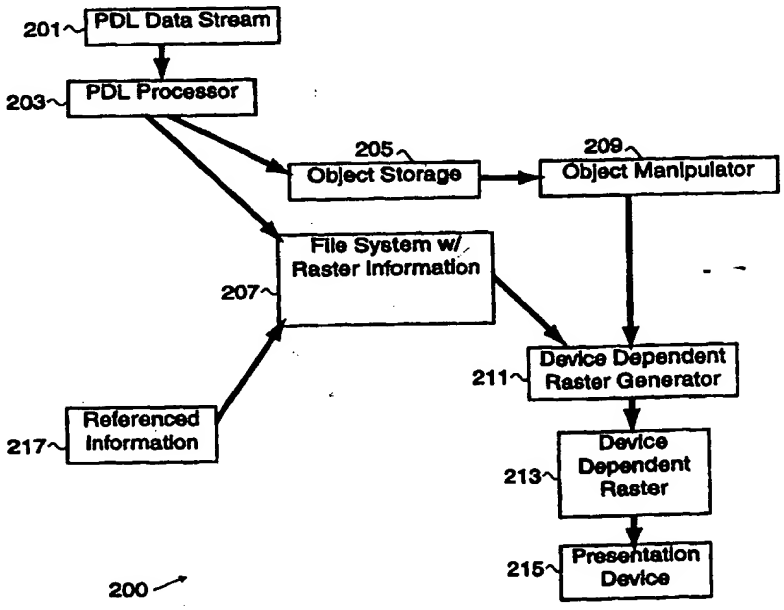
(81) Designated States: CA, CN, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD, APPARATUS AND COMPUTER PROGRAM PRODUCT FOR OBJECT-BASED RASTER IMAGING

(57) Abstract

The invention improves the performance of systems that process page description languages. The invention generates objects that represent graphical aspects of a visually perceivable image. These objects can be rasterized in parallel to generate an object raster representation of each object. Sampled images (including hierarchical multiple-resolution sampled images) can also be separately processed and stored within the file system for repeated access. A device dependent raster is generated by assembling the contribution made by each object raster representation. One aspect of the invention allows for the sending of the document representation as a stream of objects to the presentation device (thus, off-loading the PDL processing from the raster image processor). The invention also allows for post-PDL processing manipulation of the objects (for example, to reference a high resolution version of a sampled image instead of a low resolution version of the image). Another aspect of the invention reduces the number of generated graphical objects by maintaining PDL objects that contain PDL information along with image objects that contain sampled image data. These two aspects of the invention can be combined as appropriate for performance considerations.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

APPLICATION FOR UNITED STATES LETTERS PATENT FOR

A

METHOD, APPARATUS & COMPUTER PROGRAM PRODUCT FOR OBJECT-BASED RASTER IMAGING

Assignee: Pixo Arts Corporation, a corporation having an office at 2570 W. El Camino Real, Suite 110, Mountain View, CA 94040

INVENTORS	Citizenship	Residence
Liu, Clive K.	USA	Los Altos, CA, USA
Vasilev, Dmitriy N.	USA	Saratoga, CA, USA
Singh, Rarendra P.	India	New Delhi, India

A METHOD, APPARATUS & COMPUTER PROGRAM PRODUCT FOR OBJECT-BASED RASTER IMAGING

This application claims priority under 35 USC 119(e) of co-pending provisional application number 60/062,348 filed 10/15/97 entitled *Object-Based Raster Imaging Processing of Color Documents,* hereby incorporated by reference in its entirety.

Background of the Invention

Field of the Invention

This invention relates to the field of raster image generation for presentation devices such as printer or display devices.

Background

The prior art of raster image processing for generating data for a presentation device (such as a printer) is well described in Figs. 1 through 6 of the parent application.

The prior art has difficulty processing high-resolution color images because of the large amounts of memory required to assemble the raster image prior to its being sent to the presentation device. These memory requirements increase the cost of the raster image processor (RIP) used to generate the device dependent raster used by the presentation device. The prior art also has difficulty processing high-resolution color images because the document size is limited by the memory size of the raster image processor used to process the page.

Another problem with the raster image processing of page description languages (PDLs) (such as the PostScript® PDL and portable document format (PDF) languages) is that a powerful computer processor is needed to timely process these languages. Thus, these processors are expensive.

PDLs are page oriented in that the PDL instructions that define the page are interrelated and graphical aspects of one portion of the page can depend on another portion of the page. Thus, it is very difficult to parallel process these PDLs in units smaller than a page. Therefore, input streams that generate single page images can not take advantage of standard parallel processing techniques (as these techniques parallel process multiple pages, not multiple portions of a page). For these reasons, the initiation of the presentation process of a

complex page is delayed because only a single processor is used to prepare the page for rendering.

The PostScript PDL is further described in the *PostScript Language Reference Manual*, 2nd edition, © 1990 Adobe Systems Inc., ISBN 0-201-18127-4, Addison-Wesley Publishing Company. The PDF language is further described in the *Portable Document Format Reference Manual*, version 1.2, © 1996 by Adobe Systems Incorporated, ISBN 0-201-62628-4.

Another problem is that of processing of high-resolution sampled images in a PDL processor. In order to accommodate different resolution presentation devices, the PDL data stream includes sampled images having a resolution that is compatible with the higher resolution presentation device. For example, if a sampled image is included in the PDL data stream, the PDL processor converts the sampled image information into a presentation device specific format for each use of the sampled image. This conversion is a high-overhead operation. When presenting the PDL data stream on multiple presentation devices, the PDL processor processes the sampled image for each device (possibly adjusting the resolution of the sampled image to match the device). Additionally, the PDL data stream is often presented on computer monitors multiple times and possibly with differing resolutions (depending on the magnification of the image being displayed). Each presentation of the PDL data stream results in the sampled image being processed by the PDL processor.

Another problem with the known art is the existence of differing form-factors for different presentation devices (such as a computer monitor and printer). In this case the PDL is separately processed and a device dependent raster is separately generated for each type of output device because the known art does not include any technique to reposition or scale the graphical aspects of the page described by the PDL after the PDL is created.

It would be advantageous to provide a system that conserves memory usage when generating an image, that can process large PDL flies beyond the memory limit of the raster image processor or processor architecture, that allows for parallel processing of a page, that can use less expensive processors, allows manipulation of graphical objects generated by the PDL processor and that can bypass PDL processing of sampled images by the PDL processor.

Summary of the Invention

The invention provides object-oriented systems, apparatus, methods, and program products for processing images. Objects that represent imaging aspects of a document representation are described by a page description language (PDL). These objects can be separately rasterized and the resulting object raster representations can be stored (such as in a file system). A device dependent raster is generated by assembling the contribution made by each object raster representation. The invention enables parallel processing on an object-by-object basis, allows for the transmission of the document representation as a stream of objects and/or partial object raster representations to the presentation device (thus, bypassing portions of the processing of sampled images by the PDL processor or off-loading the PDL processing from the raster image processor).

Another aspect of the invention separates sampled images from a PDL data stream (such as Postscript). These sampled images are stored in a file system and accessed using a data structure (or object-oriented programming object). The PDL portion of the data stream is processed separately from the sampled images. This capability allows different images (including different resolution images and completely different images) to be stored in the file system. The stored sampled images are used with the raster information represented by the PDL data stream to generate the device dependent raster. Once the sampled images are stored on the file system, the sampled images need not be included with subsequent sendings of the PDL data stream. This improves the performance of the raster image processor and reduces the time to send the data stream to the raster image processor. The PDL data stream can be represented by a PDL object that contains the PDL data for subsequent rasterization. The PDL object itself can reference an object raster representation of the PDL data stored in the PDL object (like sampled images, the PDL's object raster representation can be stored at multiple resolutions).

Description of the Drawings

Fig. 1 illustrates a conceptual overview of a preferred embodiment the invention;

Fig. 2 illustrates a structural overview of a preferred embodiment the invention;

Fig. 3 illustrates a process used in a preferred embodiment the invention;

Fig. 4A illustrates one parallel processing architecture for a preferred embodiment the invention;

Fig. 4B illustrates another parallel processing architecture of a preferred embodiment the invention;

Fig. 4C illustrates a multi-processor configuration of a preferred embodiment the invention;

Fig. 5A illustrates a document presentation process used in a preferred embodiment the invention;

Fig. 5B illustrates a process PDL procedure used in a preferred embodiment the invention;

Fig. 5C illustrates a graphical object rasterization process used in a preferred embodiment the invention;

Fig. 5D illustrates a 'scalable print document data' procedure used in a preferred embodiment the invention;

Fig. 6 illustrates a graphical object data structure used in a preferred embodiment the invention;

Fig. 7A illustrates a process used to assemble a device raster by a preferred embodiment the invention;

Fig. 7B details the 'process boundary' procedure 703 shown in Fig. 7A; and

Fig. 7C details an alternative version of the 'process boundary' procedure 703 shown in Fig. 7A.

Description of the Preferred Embodiments

Notations and Nomenclature

The following notations and nomenclature are provided to assist in the understanding of the present invention and the preferred embodiments thereof.

Data Structure — A data structure is an ordered arrangement of storage in memory for variables. A data structure is often part of an object oriented programming object.

Device Dependent Raster — A device dependent raster is the raster information that can be used by a presentation device to display a visual image.

Object — An object in the object oriented programming paradigm is an association between programmed methods and the data structures defined by a class and the instantiated storage that represents an object of the class.

Page — A page is information that is visually presented. For example, but without limitation, a page produced by a printing device corresponds to a page of paper holding markings; and a page produced by a computer display monitor is the image produced by the monitor. Thus, a page is any visual image generated in a perceivable form. A page object is an object-oriented programming object that represents a page.

Page Description Language — A page description language (PDL) is information (data and/or operators) that describes a visual image. The invention contemplates both page programming languages (such as the Postscript PDL that provide programming capabilities) and page description languages (such as many display terminal commands). Other well-known PDLs include (without limitation) the hypertext markup language (HTML), HPGL, and the Standard Generalized Markup Language (SGML).

Procedure — A procedure is a sequence of computerized steps that lead to a desired result. These steps are defined by one or more computer instructions. These steps are performed by a computer executing the instructions that define the steps. Thus, the term "procedure" can refer, without limitation, to a sequence of instructions, a sequence of instructions organized within a programmed-procedure, programmed-function, or programmed-method; or a sequence of instructions organized within programmed-processes executing in one or more computers.

Sampled Image — A sampled image is composed of data that represent brightness and (possibly) color information for each pixel in the sampled image.

Detailed Description

A preferred embodiment is implemented using object-oriented programming techniques. Object-oriented programming (OOP) languages associate an object's data with programmed-methods for operating on that object's data. Often, OOP objects are instantiated in a heap memory area and are based on classes that reference the programmed-methods for the OOP object. Instantiated OOP objects contain data (in instance variables) specific to that

particular instantiated OOP object. Conceptually, an OOP object contains object-related information (such as the number of instance variables in the object), the instance variables, and addresses of programmed-methods that access and/or manipulate the contents of the instance variables in the object. However, because objects often share programmed-methods and object-related information, this shared information is often extracted into a class. Thus, the instantiated object simply contains its instance variables and a pointer to its class.

Smalltalk, Java and C++ are examples of OOP languages. Smalltalk was developed in the Learning Research Group at Xerox's Palo Alto Research Center (PARC) in the early 1970s. C++ was developed by Bjarne Stroustrup at the AT&T Bell Laboratories in 1983 as an extension of the C programming language. Java is an OOP language with elements from C and C++ and includes highly tuned libraries for the Internet environment. It was developed at Sun Microsystems and released in 1995.

Further information about OOP concepts can be found in *Not Just Java* by Peter van der Linden, Sun Microsystems Press/Prentice Hall PTR Corp., Upper Saddle River, NJ, (1997), ISBN 0-13-864638-4, pages 136–149.

Fig. 1 is a conceptual overview, indicated by general reference character **100**, of some of the innovative aspects used in the invention. A document representation **101**, which has been prepared by one aspect of the invention, contains one or more page objects (such as a first page object **103**, a second page object **105**, and a third page object **107**). One or more of these page objects **103**, **105**, **107** intersect graphical objects **109** that represent information that will be presented by a presentation device (such as a computer monitor, a printer or other similar presentation device). A page object can also reference overlapping graphical objects **111**. The overlapping graphical objects **111** (and the other graphical objects **109**) can be transparent. An off-page graphical object **113** can be placed partially or completely off a page object. The page objects **103**, **105**, **107** represent a page that is capable of being rendered and defines which of the graphical objects **109**, **111**, **113** are rendered for that particular page. As will be subsequently described, object raster representations of each of the graphical objects are stored in a file system (not shown).

One type of graphical object is a PDL object. A PDL object contains PDL information (for example, information conforming to the Postscript language). In one embodiment, the PDL object does not contain any sampled image data. In an alternative embodiment, the PDL object initially does not contain sampled image data, but the process of assembling the output

raster buffer inserts portions of the sampled image data into the PDL object. The object raster representation for the PDL object can be stored at multiple resolutions to improve performance for presentation devices that allow magnification (for example (but without limitation), a computer display monitor). The multiple-resolution object raster representation is stored using techniques similar to the techniques used to store sampled images at multiple resolutions.

PDL objects are used to reduce the number of graphical objects that are stored in the file system.

The document representation **101** is rendered by storing a device dependent raster within an output buffer **115**. The device dependent raster is generated (by one aspect of the invention) by specifying a boundary **117** that intersects the objects in the document representation **101**. Contributions from the object raster representations of the intersecting objects are assembled in the output buffer **115**. The output buffer **115** is sent to a presentation device (not shown) that, in the case of a printer, presents a page **119** containing a partially presented image **121**. As the boundary **117** is advanced, the entire page is imaged. The presentation device can include (for example, but without limitation) a printer, a computer monitor, and a camera. These devices have different resolutions and different usage patterns. For example when the presentation device is a printer device, the document representation **101** is generally printed at the printer's resolution. On the other hand, when the presentation device is a computer monitor, the document representation **101** is often processed at different levels of detail (for example, when a user changes the magnification of a presented image).

In a preferred embodiment, the document representation **101** is a scalable print document (SPD). The SPD is a collection of objects (subsequently described with respect to Fig. 6) that embody the document representation **101**.

Fig. 2 illustrates a structural overview, indicated by general reference character **200**, that indicates the structure of the invention. An input data stream **201** is supplied to a PDL processor **203** through a network or input/output port from a computer (not shown). The PDL processor **203** uses the input data stream **201** to generate objects (in an object storage **205**) that represent imaging aspects of a document representation. These imaging aspects are defined by the data within the input data stream **201**. The graphical objects in the object storage **205** are rasterized (as is subsequently described with regards to Fig. 5C, Fig. 7B, and Fig. 7C) and these object raster representations can be stored in a file system **207** or other storage

mechanism. One embodiment includes an object manipulator **209** that can be used to modify the objects in the object storage **205**. The objects in the object storage **205** and the object raster representations stored in the file system **207** are used by a device dependent raster generator **211** to generate device dependent raster data **213**. The device dependent raster data **213** is sent to a presentation device **215**. The presentation device **215** (the device raster process mechanism) uses the device dependent raster data **213** to create an image that can be presented visually.

The objects generated by the PDL processor **203** can be in SPD format and can be stored in the object storage **205** or on a computer system's file-storage. As is subsequently described with respect to Fig. 5A, the PDL processor **203** can also accept a data stream in the SPD format.

Referenced information **217** (such as a very high-resolution sampled image) can also be directly stored within the file system **207** or extracted by the PDL processor **203** from the input data stream **201**. The referenced information **217** can be a hierarchical sampled image (such as an image in FlashPix format). This allows the sampled image to be displayed on different presentation devices at different resolutions with minimal processing to resize the sampled image. Thus, the device dependent raster generator **211** can select a sampled image that resides in the file system having a resolution that best matches the resolution of the presentation device. The sampled image can also be accessed on the file system using the open prepress interface (OPI), which defines the location of the sampled image.

The object manipulator **209** can modify the contents of the object's fields such as the object's coordinates, layer, transparency and selection of raster information for sampled images. These modifications effect the operation of the device dependent raster generator **211** in generating the device dependent raster.

The input data stream **201** includes information that can be processed by the PDL processor **203**. For example this information can conform to (without limitation): the Postscript PDL including embedded sampled images, the open prepress interface (OPI) variant of Postscript, the portable document format (PDF) including embedded sampled images, the OPI variant of the portable document format, the Hewlett-Packard graphics language (HPGL), and the hypertext markup language (HTML) format including sampled images (such as JPEG and GIF images). In addition, the input data stream **201** can also consist of scalable print document (SPD) data.

One skilled in the art will understand that the term "file system" includes any file storage system including memory resident file systems as well as other file storage systems (including, without limitation, magnetic, optical, or other system for storing information).

Fig. 3 illustrates a process, indicated by general reference character **300**, that summarizes the method of a preferred embodiment. The process **300** initiates at a 'start' terminal **301** and continues to a 'receive data stream' step **303**. The 'receive data stream' step **303** receives the data stream that defines the document representation. The data stream is consumed and processed by a 'generate objects' step **305** that generates the page object, the graphical objects and the document representation object shown in Fig. 1. Next, a 'rasterize graphical objects' step **307** rasterizes the graphical objects that intersect a page object. The object raster representations, thus created, are stored in a file system by a 'store graphical object rasters in file system' step **309**. The objects and object raster representations are used by an 'assemble device dependent raster' step **311** to generate a device dependent raster that is sent to a presentation device. A 'present device dependent raster' step **313** causes the presentation device to visually present this information defined by the data stream and embodied in the device dependent raster. The process completes at an 'end' terminal **315**. Details of the inventive steps above are subsequently described.

Fig. 4A illustrates a parallel processing architecture, indicated by general reference character **400**, used by a preferred embodiment. Although other preferred embodiments need not use parallel processing, the invention's creation of objects enables parallel processing for rasterization of the graphical objects and for raster processing of the resulting object raster representations. In such an embodiment, a PDL data stream **401** is received by a PDL processor **403** that creates object representations **405** representing imaging aspects of the document representation created by the PDL data stream **401** (the PDL processor **403** generates data in a SPD format). Some of the graphical objects are then rasterized by a 'first object rasterization' process **407** and the resulting object raster representations further processed by a 'first raster processing' process **409**. The 'first raster processing' process **409** often includes anti-aliasing, color processing and screening processes. The resulting object raster representations are then assembled by a 'device raster assembly' process **411** to generate a device dependent raster. The device dependent raster is sent to a presentation device that performs an image presentation step **413**. Parallel processing can occur during the generation of the object raster representations and the raster processing steps as shown by a 'second object rasterization' process **415** and a 'second raster processing' process **417**. Parallel

processing is accomplished by allocating the processing of the objects to multiple threads (sending the objects to multiple threads-of-execution). The threads can be executed by multiple processors or within a single processor. The multiple processors can be connected by a message mechanism (such as an inter-processor communication mechanism or a network in a networked client/server computer system).

Fig. 4B illustrates another parallel processing architecture, indicated by general reference character 430, for parallel processing PDL objects and image objects. In the following description, the PDL data is Postscript data. A PDL data stream 431 is supplied to a PDL preprocessor 433. The PDL preprocessor 433 parses the PDL data stream 431 to separate sampled images included in the PDL data stream 431 from the Postscript information. The PDL preprocessor 433 generates a PDL object 435 that contains the Postscript information for a page object defined by the PDL data stream 401. The PDL object 435 is processed by a Postscript processor 437 to generate a corresponding object raster representation. The PDL preprocessor 433 also generates an image object 439 that contains information that represents a sampled image. The image object 439 is processed by an image processor 441. The Postscript processor 437 and the image processor 441 can be processed in parallel. A 'device raster assembly' process 443 assembles the device dependent raster using the object raster representation generated from the PDL object 435 by the Postscript processor 437, and generated from the image object 439 by the image processor 441. The device dependent raster is sent to a presentation device that performs an image presentation step 445.

One aspect of the invention provides mechanisms to distribute processing of an input stream. Fig. 4C illustrates a multi-processor configuration, indicated by general reference character 450, that includes a first processor 451 that sends objects to a second processor 453 over an inter-processor communication mechanism 455. The first processor 451 generates the objects by processing an input data stream (not shown). The second processor 453 receives the objects and generates the object raster representation for at least some of the objects. Additional processors (not shown) can also be used to parallel process the objects and the object raster representations. The inter-processor communication mechanism 455 can be a network (in which case the first processor 451 and the second processor 453 could, for example, be in a client-server relationship) or a mechanism that supports multiple processors in the same computer. The multi-processor configuration 450 also can include a removable media reader 457 that is made available to either or both the first processor 451 and the second processor 453. The removable media reader 457 is able to read a removable media 459 that

contains program code and data **461**. The program code and data **461** when executed by one of the processors causes the processors to perform the inventive steps.

The invention distinguishes between different types of data streams. **Fig. 5A** illustrates a document presentation process, indicated by general reference character **500**, for determining how to process input data to generate the document representation **101** described in Fig. 1. The document presentation process **500** initiates at a 'start' terminal **501** and continues to a 'select data type' procedure **503**. The 'select data type' procedure **503** examines the data stream and determines what type of data is contained in the data stream. This determination uses methods well understood in the art. Based on this determination, the data is sent to a 'process PDL data' procedure **505**, or a 'process scalable print document data' procedure **507**. Each of these procedures is subsequently described. Each of these procedures generates a device dependent raster representing the information that will be presented by the presentation device. The device dependent raster is sent to the presentation device by a 'send raster to device' procedure **511**. Once the device dependent raster is sent to the presentation device, the document presentation process **500** completes through an 'end' terminal **513**.

The 'process PDL data' procedure **505** represents any PDL processor (other than the 'process scalable print document data' procedure **507**) that consumes and processes PDL data. The PDL processor includes (without limitation) those that process Postscript, HTML, HPGL, and PDF data.

In a preferred embodiment, the 'generate objects from PDL' procedure **523** generates graphical objects for each graphical aspect that is to be presented. In another preferred embodiment, used to reduce the total number of generated graphical objects, the 'generate objects from PDL' procedure **523** generates a PDL object that (for example, without limitation) contains the Postscript information that describes the graphical aspects of the page (except for the sampled images) and image objects that represent the sampled images. In both cases, the PDL processor does not process the information represented by the image objects. Instead the PDL processor identifies the sampled images and generates objects representing these images for later processing. Thus, the 'generate objects from PDL' procedure **523** bypasses the processing of the sampled images. The sampled images are eventually stored in the file system (possibly at different resolutions (for example, using the FlashPix format)) as is subsequently described.

Fig. 5B illustrates a 'process PDL data stream' procedure, indicated by general reference character **520**, for processing a data stream that consists of a page description language (for example, the Postscript PDL). The 'process PDL data stream' procedure **520** is invoked at the 'process PDL data' procedure **505** of Fig. 5A and initiates at a 'start' terminal **521**. Then the 'process PDL data stream' procedure **520** continues to a 'generate objects from PDL' procedure **523** that generates the objects that represent imaging aspects of the document representation (instead of directly generating a device dependent raster). The 'generate objects from PDL' procedure **523** also stores object raster representations in a file system. The object raster representation is a raster representation of the imaging aspect of the graphical object. When the page is presented, the device dependent raster is generated by assembling the raster representations of the graphical objects that intersect the page object being presented. Certain aspects of the 'generate objects from PDL' procedure **523** are subsequently described with respect to Fig. 5C. The 'process PDL data stream' procedure **520** completes through an 'end' terminal **525**.

The 'process PDL data stream' procedure **520** can bypass processing of a sampled image. It does so by storing the sampled image data in the file system and by associating that data with a rasterized graphical object. The sampled image data can also be directly downloaded to and stored in the file system (possibly with multiple resolutions) external to the PDL processing. Thus, the 'process PDL data stream' procedure **520** need not process the sampled image data and can instead generate a placeholder for the sampled image and reference the downloaded sampled image data.

Fig. 5C illustrates a 'graphical object rasterization' process **530** that can be invoked by the 'generate objects from PDL' procedure **523** of Fig. 5B for each page object in the document representation. The 'graphical object rasterization' process **530** initiates at a 'start' terminal **531** and continues to an 'iterate each graphical object within page object' procedure **533**. The 'iterate each graphical object within page object' procedure **533** locates each graphical object that intersects the current page object. When all graphical objects that intersect the page object have been located and processed, the 'graphical object rasterization' process **530** continues to an 'assemble device raster' procedure **535**. The 'assemble device raster' procedure **535** generates the device dependent raster for the page using the object raster representations and the graphical objects as is subsequently described with respect to Fig. 7A and, Fig. 7B or Fig. 7C. The 'graphical object rasterization' process **530** completes through an 'end' terminal **537**.

Each graphical object that intersects the page object at the 'iterate each graphical object within page object' procedure **533** is examined by a 'PDL object' decision procedure **538** that determines whether the object contains Postscript information. If the object is a PDL object, the 'graphical object rasterization' process **530** continues to the 'iterate each graphical object within page object' procedure **533** to iterate subsequent objects. The PDL object will be subsequently rasterized as subsequently shown in Fig. 7B or Fig. 7C.

However, if the object is not a PDL object, the 'graphical object rasterization' process **530** continues to an 'object rasterized' decision procedure **539** to determine whether the graphical object is already rasterized. Examples of an already rasterized graphical object include (without limitation) a sampled image and a bitmapped glyph. If the graphical object is not already rasterized, the 'graphical object rasterization' process **530** continues to a 'rasterize graphical object' procedure **541** that generates an object raster representation for the graphical object. Thus, if the graphical object defines a shape within a bounding box, the 'rasterize graphical object' procedure **541** generates a raster representation of the shape. One skilled in the art will understand that optimizations to reduce the amount of memory needed to store the raster for the bounding box (such as decomposing the shape into graphical objects containing smaller segments of the shape) can be applied prior to, or as part of, the 'rasterize graphical object' procedure **541**. Once the object raster representation is created, it is saved in the file system by a 'save graphical object raster' procedure **543**. Once the graphical object raster is saved, the 'graphical object rasterization' process **530** continues to the 'iterate each graphical object within page object' procedure **533** to process the next graphical object that intersects the page object.

However, if at the 'object rasterized' decision procedure **539** the graphical object is determined to already contain an object raster representation, the 'graphical object rasterization' process **530** continues to a 'straight presentation' decision procedure **545**. The 'graphical object rasterization' process **530** examines the graphical object to determine whether the existing object raster representation is to be presented or whether a new object raster representation is to be generated at a different resolution. If the object raster representation is to be used at its existing resolution, the 'graphical object rasterization' process **530** continues to the 'save graphical object raster' procedure **543** for storing the provided raster information in the file system (as has been previously described). Otherwise, a 'generate hierarchical image resolutions' procedure **547** uses techniques well understood in the art to processes the raster data in the graphical object to generate a hierarchical set of raster

images (or image tiles) of differing resolutions for the sampled image. A preferred embodiment stores these differing resolutions in the FlashPix format.

After completion of the 'graphical object rasterization' process **530**, an object raster representation for each object (other than a PDL object) that intersects the page has been stored in the file system.

One aspect of the invention is that the PDL processing mechanism (such as the 'generate objects from PDL' procedure **523**) need not process the sampled image included within the data stream. Instead, the PDL processing mechanism bypasses the processing of the sampled image and stores the sampled image in the file system. Once stored in the file system, the 'assemble device raster' procedure **535** can use the sampled image to assemble the device dependent raster. This capability allows the stored sampled image to be replaced by another sampled image of different resolution or content without requiring that the data stream be reprocessed by the 'generate objects from PDL' procedure **523**.

Fig. 5D illustrates a 'scalable print document data' process, indicated by general reference character **570** that is invoked by the 'process scalable print document data' procedure **507** of Fig. 5A. The 'scalable print document data' process **570** initiates at a 'start' terminal **571** and continues to a 'straight presentation' decision procedure **573** that determines whether the SPD data stream contains raster data at the same resolution as the presentation device. If not, the 'scalable print document data' process **570** continues to a 'graphical object rasterization' procedure **575** that invokes the 'graphical object rasterization' process **530** of Fig. 5C to re-generate the object raster representation for each non-PDL object in the SPD data stream. The re-generated object raster representation is stored in the file system. This process also can generate different resolutions of sampled images that are included in the data stream or already resident in the file system.

However, if the 'straight presentation' decision procedure **573** determines that the data stream contains object raster representations at the same resolution as the presentation device, the 'scalable print document data' process **570** continues to the 'save object raster representation' procedure **577** to store the non-PDL objects' object raster representations in the file system. After the 'graphical object rasterization' procedure **575** or the 'save object raster representation' procedure **577** the 'scalable print document data' process **570** continues to an 'assemble device raster' procedure **579** that creates the device dependent raster from the stored raster as is subsequently described with respect to Fig. 7A and, Fig. 7B or Fig. 7C.

Then, the 'scalable print document data' process **570** completes through an 'end' terminal **581**. The PDL objects are rasterized by the 'assemble device raster' procedure **579** as described in relation to Fig. 7B or Fig. 7C.

One skilled in the art will understand that the 'assemble device raster' procedure **535**, and the 'assemble device raster' procedure **579** can be the same programmed-procedure.

Fig. 6 illustrates a graphical object data structure, indicated by general reference character **600**, used to represent the graphical objects in the document representation. The graphical object data structure **600** includes an 'object coordinates' field **601** that contains or references the coordinates of the graphical object in the document representation. These coordinates often include a page number, a horizontal position and vertical position for the graphical object's bounding box with respect to the specified page. Other coordinates can be used to represent the position of the graphical object. A 'bounding box' field **603** contains or references the bounding box for the image represented by the graphical object. The 'bounding box' field **603** is used with the boundary **117** (as indicated in Fig. 1) and the page object to determine the intersection of the graphical object, the page object and the boundary **117**. A 'rasterized' field **605** contains or references a boolean value that indicates whether an object raster representation exists for the graphical object. If the 'rasterized' field **605** value is TRUE, a 'raster location' field **607** contains or references a file system specification that specifies the location of the object raster representation in the file system. A 'layer' field **609** is used to determine which graphical object takes precedence over another graphical object when two or more graphical objects overlap during the assembly of the device dependent raster. The 'transparency' field **611** contains or references transparency information for the graphical object. This transparency information is used by well-known functions during the assembly of the device dependent raster. The graphical object data structure **600** also includes 'object dependent information' fields **613** that contain information about the graphical object. This information is often used to generate the object raster representation. For example, the 'object dependent information' fields **613** can contain Postscript information for a PDL object or a reference to the original data of a sampled image that has been magnified (the magnified sampled data image is stored as indicated by the 'raster location' field **607**), or a reference to a replacement sampled image associated with the image object.

The information stored in these fields can be modified by an external program (such as the object manipulator **209** shown in Fig. 2). For example, an external program can reposition the graphical objects by modifying the values in the 'object coordinates' field **601**, or change

the resolution of a sampled image by changing the value in the 'raster location' field **607**. In addition, the referenced information **217** can be downloaded to the file system. In this circumstance, the downloaded information is accessible to the system without requiring that the downloaded information be included within the input data stream **201** every time it is processed. This allows commonly-used information (such as sampled images or PDL objects) to be locally stored instead of being sent with the data stream every time the data stream is processed.

Fig. 7A illustrates an 'assemble device raster' process, indicated by general reference character **700**, used to assemble the device dependent raster for the presentation device. The 'assemble device raster' process **700** initiates at a 'start' terminal **701** and continues to a 'process boundary' procedure **703** that assembles the object raster representations stored within the file system into the device dependent raster for the presentation device. The 'process boundary' procedure **703** is subsequently described with respect to Fig. 7B. After the device dependent raster for the current boundary is generated within the output buffer, the output buffer is sent to the presentation device by a 'send output buffer' procedure **705**. The presentation device processes the object raster representation and presents the visible image. Next, an 'end of page' decision procedure **707** determines whether the current boundary completed the page object. If the current boundary did not complete the page object, the 'assemble device raster' process **700** continues to an 'advance boundary' procedure **709** that advances the boundary. Once the boundary is advanced, the 'assemble device raster' process **700** continues to the 'process boundary' procedure **703** to process graphical objects that intersect the advanced boundary.

However, if the 'end of page' decision procedure **707** determines that the current boundary is at the end of the page object, the 'assemble device raster' process **700** continues to an 'end of page processing' procedure **711**. The 'end of page processing' procedure **711** causes the presentation device to complete the presentation process for the rendered page image. Next, the 'assemble device raster' process **700** continues to an 'end of document' decision procedure **713** that determines whether the last page of the document representation has been rendered. If so, the 'assemble device raster' process **700** completes through an 'end' terminal **715**. Otherwise, the 'assemble device raster' process **700** continues to a 'start of page processing' procedure **717** that prepares the presentation device for receiving device dependent raster for a new page. The 'assemble device raster' process **700** then resets the boundary to the top of the next page object at a 'reset boundary' procedure **719** and continues

to the 'process boundary' procedure **703** to generate the device dependent raster for the new page object.

Fig. 7B illustrates an 'assemble boundary' process, indicated by general reference character **730**, that assembles the raster contribution from the graphical objects that intersect the boundary to generate a device dependent raster in the output buffer. The process **730** can be invoked by the 'process boundary' procedure **703** of Fig. 7A and initiates at a 'start' terminal **731**. The process **730** continues to an 'iterate each color plane' procedure **733** that uses color information in the graphical objects to generate the device dependent raster suitable for a color presentation device. Once all relevant color planes have been processed, the process **730** completes through an 'end' terminal **735**. The color plane processing is dependent on the presentation device and the characteristics of the graphical objects in the document representation (that is, black and white documents can also be rendered).

For each color plane, a 'boundary intersects PDL object' decision procedure **736** determines whether the boundary intersects a PDL object. If not, the process **730** continues to an 'iterate each area in boundary' procedure **737**. Otherwise, the process **730** continues to a 'process PDL contribution' procedure **738** that passes the PDL information within the PDL object to a PDL processor along with the area defined by the boundary (for example, a Postscript processor processes the PDL object if it contains Postscript information). The PDL processor then generates an object raster representation of the information that contributes to the boundary. This object raster representation can be stored in the file system or used to initialize the output buffer. In addition, the 'process PDL contribution' procedure **738** can store the object raster representation for the PDL object as a hierarchical image. Next, the process **730** continues to the 'iterate each area in boundary' procedure **737**.

The 'iterate each area in boundary' procedure **737** processes each relevant area within the boundary. An area can be a raster line, a tile, or other appropriate area. As each area in the boundary is iterated by the 'iterate each area in boundary' procedure **737**, an 'iterate each object intersecting area' procedure **739** iterates each graphical object having a bounding box that intersects the iterated area. An 'obtain object contribution to area' procedure **741** extracts portions of the raster data from each graphical object's object raster representation that intersects the iterated area.

A 'PDL object' decision procedure **742** determines whether the extracted raster information has been fully processed (for example, Postscript raster generation procedures can

generate anti-aliased raster information). If the extracted raster information has been fully processed the process **730** continues to the 'iterate each object intersecting area' procedure **739** to process the next intersecting object. Otherwise the raster extracted from the object raster representation is processed by an 'anti-alias processing' procedure **743** that performs well-known anti-aliasing processing, a 'color processing' procedure **745** that performs well-known color processing, and a 'screen processing' procedure **747** that performs well-known screen processing. When all of the raster contributions for the area have been assembled, a 'save area in output buffer' procedure **749** stores the area in the output buffer. Then the process **730** returns to the 'iterate each area in boundary' procedure **737** to iterate the next area in the boundary. When all areas in the boundary have been processed, the process **730** continues to the 'iterate each color plane' procedure **733** for subsequent color plane processing or completion.

Fig. 7C illustrates an alternative 'assemble boundary' process, indicated by general reference character **750**, that also can be used to assemble the raster contribution from the objects that intersect the boundary to generate a device dependent raster in the output buffer. The alternative 'assemble boundary' process **750** can be invoked by the 'process boundary' procedure **703** of Fig. 7A and initiates at a 'start' terminal **751**. The alternative 'assemble boundary' process **750** continues to an 'iterate PDL object' procedure **753** that iterates each PDL object that intersects the boundary. When all PDL objects have been iterated the alternative 'assemble boundary' process **750** completes through an 'end' terminal **755**.

For each iterated PDL object, an 'iterate intersecting sampled image' procedure **757** then iterates each sampled image that also intersects the boundary. An 'insert sampled image data into PDL object' procedure **759** then extracts portions of the sampled image data intersecting the boundary from the sampled image object and inserts this data into the PDL object. This continues for each sampled image object iterated by the 'iterate intersecting sampled image' procedure **757**. Once all the intersecting sampled image objects have been iterated, the alternative 'assemble boundary' process **750** continues to a 'process PDL object for boundary' procedure **761**. The 'process PDL object for boundary' procedure **761** passes the PDL object to the PDL processor (not shown). The PDL processor operates on the PDL and sampled image data within the PDL object to generate the device dependent raster for the boundary. A 'save raster in output buffer' procedure **763** stores the device dependent raster (created by the 'process PDL object for boundary' procedure **761**) in the output buffer. Then the alternative 'assemble boundary' process **750** continues to the 'iterate PDL object'

procedure **753** to process other PDL objects. In addition, the 'process PDL object for boundary' procedure **761** can store the object raster representation for the PDL object intersecting the boundary as a hierarchical multi-resolution image.

In a first preferred embodiment of the process shown in Fig. 7C, a single PDL object contains the entire PDL description for a page. A second preferred embodiment of the process shown in Fig. 7C uses a plurality of non-intersecting PDL objects to define a page.

One skilled in the art will understand that the computers represented in Figs. 1A, 1B, and 1C along with the raster image processors of Figs. 3, 4, and 5 (all of the parent provisional application) and Fig. 4C are capable of loading programs from a removable media, from a network (either directly or by use of an intermediate computer and an I/O port), or from a read-only memory device. Thus, the invention can be a computer program product containing program code embodied in a computer readable media or transmitted over a carrier wave.

One skilled in the art will understand that the invention provides a flexible and efficient method and apparatus for generating device dependent raster data for a presentation device.

From the foregoing, it will be appreciated that the invention has (without limitation) the following advantages:

1) higher performance because of the ability to parallel process PDL data streams on an object-by-object basis instead of a page-by-page basis;

2) the ability to conserve memory resulting from storing device dependent raster data on a file system;

3) the ability to bypass processing of sampled images by a PDL processor by storing the sampled image data (possibly at multiple resolutions) on a file system;

4) the ability to process large PDL data streams that exceed the memory available to the PDL processor, the raster image processor, or the computer architecture;

5) the ability to match the resolution of a sampled image to the resolution of the presentation device so as to provide the best presented image by the presentation device with the least amount of data.

6) the ability to modify certain object parameters without reprocessing the PDL data stream.

One skilled in the art will understand that although the following description of the invention is cast within an object-oriented paradigm, the techniques disclosed are applicable to other programming paradigms. Such a one will also understand that the invention can be applied to objects and data structures other than ones used to illustrate the invention's use. Although the present invention has been described in terms of the presently preferred embodiments, various modifications and alterations can be made without departing from the scope of the invention. Accordingly, the scope of the invention is not to be limited to the particular invention embodiments discussed herein, but should be defined only by the appended claims and their equivalents.

Claims

What is claimed is:

1. A computer controlled method for generating a device dependent raster for a presentation device, said device dependent raster specified by a page description language (PDL), said method including steps for:

processing a data stream, that conforms to said PDL, to generate one or more objects, said one or more objects representing imaging aspects of a document representation;

storing a plurality of object raster representations in a file system, each of said plurality of object raster representations defined by at least one of said one or more objects; and

assembling said device dependent raster from portions of said plurality of object raster representations.

2. The computer controlled method of claim 1, further including processing said device dependent raster by said presentation device.

3. The computer controlled method of claim 1, wherein said one or more objects include one or more coordinate values locating said one or more objects within said document representation.

4. The computer controlled method of claim 1, wherein said data stream includes color information.

5. The computer controlled method of claim 1, wherein one of said one or more objects is a PDL object.

6. The computer controlled method of claim 5, wherein said PDL object contains Postscript information.

7. The computer controlled method of claim 5, wherein said PDL object contains information that conforms to a PDL format selected from the set of the portable document format PDL, the PCL PDL, the HPGL PDL, and the hypertext markup language PDL.

8. The computer controlled method of claim 1, further including sending one of said one or more objects from a first processor to a second processor to generate one of said plurality of object raster representations.

9. The computer controlled method of claim 8, wherein said first processor is connected to said second processor with an inter-processor communication mechanism and the step of sending uses said inter-processor communication mechanism to send said one or more objects.

10. The computer controlled method of claim 9, wherein said first processor is a client computer, said second processor is a server computer, and said inter-processor communication mechanism is a network.

11. The computer controlled method of claim 8, wherein the step of sending associates one of said one or more objects with at least one thread-of-execution.

12. The computer controlled method of claim 1, wherein said data stream includes a sampled image, and said method further includes steps for:

bypassing processing of said sampled image;

storing said sampled image in said file system; and

using said sampled image, by the step of assembling, to assemble said device dependent raster.

13. The computer controlled method of claim 12, wherein said sampled image is stored in said file system as a hierarchical multi-resolution sampled image.

14. The computer controlled method of claim 1, wherein each of said one or more objects contains one or more fields, and the step of storing further includes modifying the contents of said one or more fields to effect the step of assembling.

15. The computer controlled method of claim 1, wherein said PDL is selected from the set of the PostScript PDL, the portable document format PDL, the PCL PDL, the HPGL PDL, and the hypertext markup language PDL.

16. A computer controlled method for generating a device dependent raster for a presentation device, said device dependent raster specified by a page description language (PDL), said method including steps for:

processing a data stream, that conforms to said PDL, to generate one or more objects, said one or more objects representing imaging aspects of a document representation, a first of said one or more objects being a PDL object, a second of said one or more objects being a sampled image;

inserting a portion of said sampled image into said PDL object; and

processing said PDL object to generate a portion of said device dependent raster.

17. The computer controlled method of claim 16, wherein said PDL object references an object raster representation that is stored as a hierarchical multi-resolution image.

18. The computer controlled method of claim 16, wherein said sampled image is stored as a hierarchical multi-resolution sampled image.

19. An apparatus having at least one processor and at least one memory coupled to said at least one processor for generating a device dependent raster for a presentation device, said device dependent raster specified by a page description language (PDL), said apparatus includes:

a PDL processor mechanism configured to process a data stream that conforms to said PDL to generate one or more objects that represent imaging aspects of a document representation;

a storage mechanism configured to store a plurality of object raster representations in a file system, each of said plurality of object raster representations defined by at least one of said one or more objects generated by the PDL processor mechanism; and

a device raster assembly mechanism configured to assemble said device dependent raster from portions of said plurality of object raster representations stored by the storage mechanism.

20. The apparatus of claim 19, wherein said presentation device includes a device raster process mechanism configured to process said device dependent raster to generate a visual image.

21. The apparatus of claim 19, wherein said one or more objects include one or more coordinate values locating said one or more objects within said document representation.

22. The apparatus of claim 19, wherein said data stream includes color information.

23. The apparatus of claim 19, wherein one of said one or more objects is a PDL object.

24. The apparatus of claim 23, wherein said PDL object contains Postscript information.

25. The apparatus of claim 23, wherein said PDL object contains information that conforms to a PDL format selected from the set of the portable document format PDL, the PCL PDL, the HPGL PDL, and the hypertext markup language PDL.

26. The apparatus of claim 19, further including a message mechanism configured to send one of said one or more objects from a first processor to a second processor to generate one of said plurality of object raster representations.

27. The apparatus of claim 26, wherein the message mechanism is an inter-processor communication mechanism that connects said first processor to said second processor.

28. The apparatus of claim 27, wherein said first processor is a client computer, said second processor is a server computer, and said inter-processor communication mechanism is a network.

29. The apparatus of claim 26, wherein the message mechanism associates one of said one or more objects with at least one thread-of-execution.

30. The apparatus of claim 19, wherein said data stream includes a sampled image, and said apparatus further includes:

a sampled image bypass mechanism configured to bypass processing of said sampled image by the PDL processor mechanism;

a sampled image storage mechanism configured to store said sampled image in said file system; and

a sampled image assembly mechanism configured to use said sampled image in said file system, by the device raster assembly mechanism, to assemble said device dependent raster.

31. The apparatus of claim 30, wherein said sampled image is stored in said file system as a hierarchical multi-resolution sampled image.

32. The apparatus of claim 19, wherein each of said one or more objects contains one or more fields, and the storage mechanism further includes:

an object field modification mechanism configured to modify the contents of said one or more fields to effect operation of the device raster assembly mechanism.

33. The apparatus of claim 19, wherein said PDL is selected from the set of the PostScript PDL, the portable document format PDL, the PCL PDL, the HPGL PDL, and the HTML PDL.

34. An apparatus having at least one processor and at least one memory coupled to said at least one processor for generating a device dependent raster for a presentation device, said device dependent raster specified by a page description language (PDL), said apparatus includes:

a first PDL processor mechanism configured to process a data stream, that conforms to said PDL, to generate one or more objects, said one or more objects representing imaging aspects of a document representation, a first of said one or more objects being a PDL object, a second of said one or more objects containing a sampled image;

an image data selection mechanism configured to insert a portion of said sampled image into said PDL object; and

a second PDL processor mechanism configured to process said PDL object to generate a portion of said device dependent raster.

35. The apparatus of claim 34, wherein said PDL object references an object raster representation that is stored as a hierarchical multi-resolution image on a file system.

36. The apparatus of claim 34, wherein said sampled image is stored as a hierarchical multi-resolution sampled image on a file system.

37. A computer program product including:

a computer usable storage medium having computer readable code embodied therein for causing a computer to generate a device dependent raster for a presentation device, said device dependent raster specified by a page description language (PDL), said computer readable code includes:

computer readable program code configured to cause said computer to effect a PDL processor mechanism configured to process a data stream that conforms to said

PDL to generate one or more objects that represent imaging aspects of a document representation;

computer readable program code configured to cause said computer to effect a storage mechanism configured to store a plurality of object raster representations in a file system, each of said plurality of object raster representations defined by at least one of said one or more objects generated by the PDL processor mechanism; and

computer readable program code configured to cause said computer to effect a device raster assembly mechanism configured to assemble said device dependent raster from portions of said plurality of object raster representations stored by the storage mechanism.

38. The computer program product of claim 37, wherein said one or more objects include one or more coordinate values locating said one or more objects within said document representation.

39. The computer program product of claim 37, further including computer readable program code configured to cause said computer to effect a message mechanism configured to send one of said one or more objects from a first processor to a second processor to generate one of said plurality of object raster representations.

40. The computer program product of claim 39, wherein the message mechanism is an inter-processor communication mechanism that connects said first processor to said second processor.

41. The computer program product of claim 40, wherein said first processor is a client computer, said second processor is a server computer, and said inter-processor communication mechanism is a network.

42. The computer program product of claim 39, wherein the message mechanism associates one of said one or more objects with at least one thread-of-execution.

43. The computer program product of claim 37, wherein said data stream includes a sampled image, and said product further includes:

computer readable program code configured to cause said computer to effect a sampled image bypass mechanism configured to bypass processing of said sampled image by the PDL processor mechanism;

computer readable program code configured to cause said computer to effect a sampled image storage mechanism configured to store said sampled image in said file system; and

computer readable program code configured to cause said computer to effect a sampled image assembly mechanism configured to use said sampled image in said file system, by the device raster assembly mechanism, to assemble said device dependent raster.

44. The computer program product of claim 37, wherein each of said one or more objects contains one or more fields, and the storage mechanism further includes:

computer readable program code configured to cause said computer to effect an object field modification mechanism configured to modify the contents of said one or more fields to effect operation of the device raster assembly mechanism.

45. A computer program product including:

a computer usable storage medium having computer readable code embodied therein for causing a computer to generate a device dependent raster for a presentation device, said device dependent raster specified by a page description language (PDL), said computer readable code includes:

computer readable program code configured to cause said computer to effect a first PDL processor mechanism configured to process a data stream, that conforms to said PDL, to generate one or more objects, said one or more objects representing imaging aspects of a document representation, a first of said one or more objects being a PDL object, a second of said one or more objects containing a sampled image;

computer readable program code configured to cause said computer to effect an image data selection mechanism configured to insert a portion of said sampled image into said PDL object; and

computer readable program code configured to cause said computer to effect a second PDL processor mechanism configured to process said PDL object to generate a portion of said device dependent raster.

46. The computer program product of claim 45, wherein said PDL object references an object raster representation that is stored as a hierarchical multi-resolution image on a file system.

47. The computer program product of claim 45, wherein said sampled image is stored as a hierarchical multi-resolution sampled image on a file system.

48. A computer program product including:

a computer data signal embodied in a carrier wave having computer readable code embodied therein for causing a computer to generate a device dependent raster for a presentation device, said device dependent raster specified by a page description language (PDL), said computer readable code includes:

computer readable program code configured to cause said computer to effect a PDL processor mechanism configured to process a data stream that conforms to said PDL to generate one or more objects that represent imaging aspects of a document representation;

computer readable program code configured to cause said computer to effect a storage mechanism configured to store a plurality of object raster representations in a file system, each of said plurality of object raster representations defined by at least one of said one or more objects; and

computer readable program code configured to cause said computer to effect a device raster assembly mechanism configured to assemble said device dependent raster from portions of said plurality of object raster representations stored by the storage mechanism.

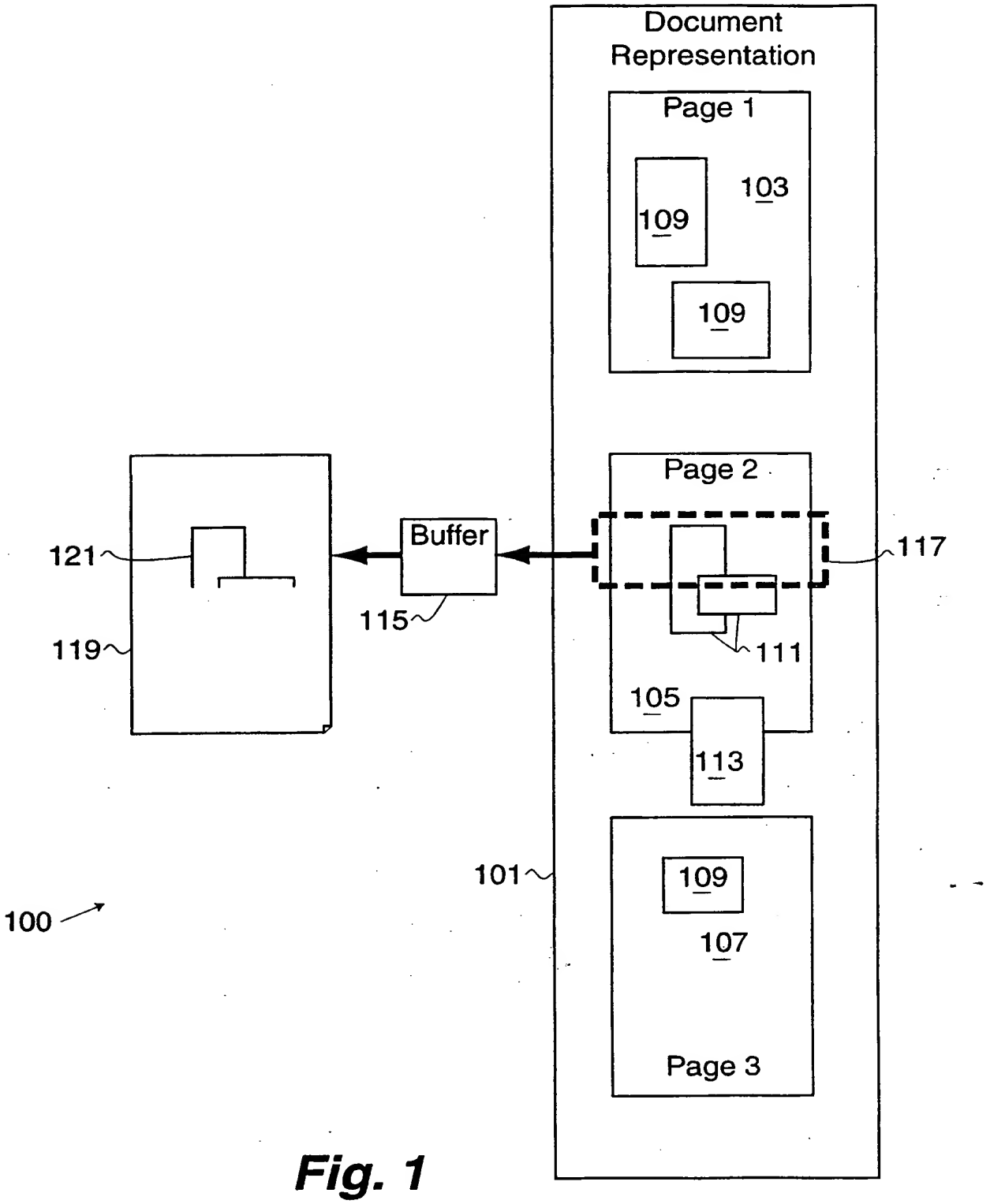

Document Representation
Page 1
103
109
109
Page 2
Buffer
121
119
115
117
111
105
113
101
109
107
Page 3
100

Fig. 1

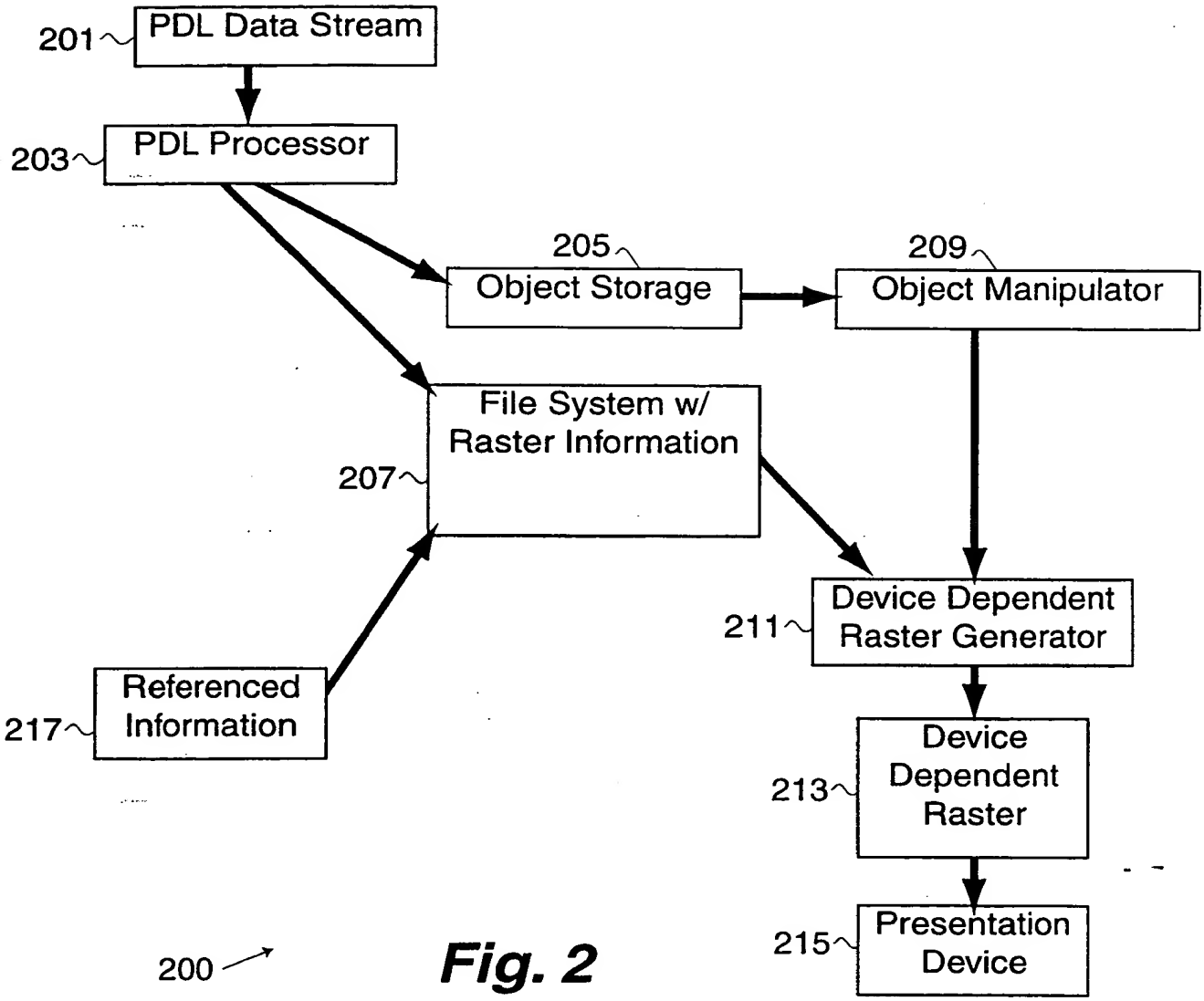

201
PDL Data Stream
203
PDL Processor
205
Object Storage
209
Object Manipulator
File System w/
Raster Information
207
211
Device Dependent
Raster Generator
217
Referenced
Information
213
Device
Dependent
Raster
215
Presentation
Device
200

Fig. 2

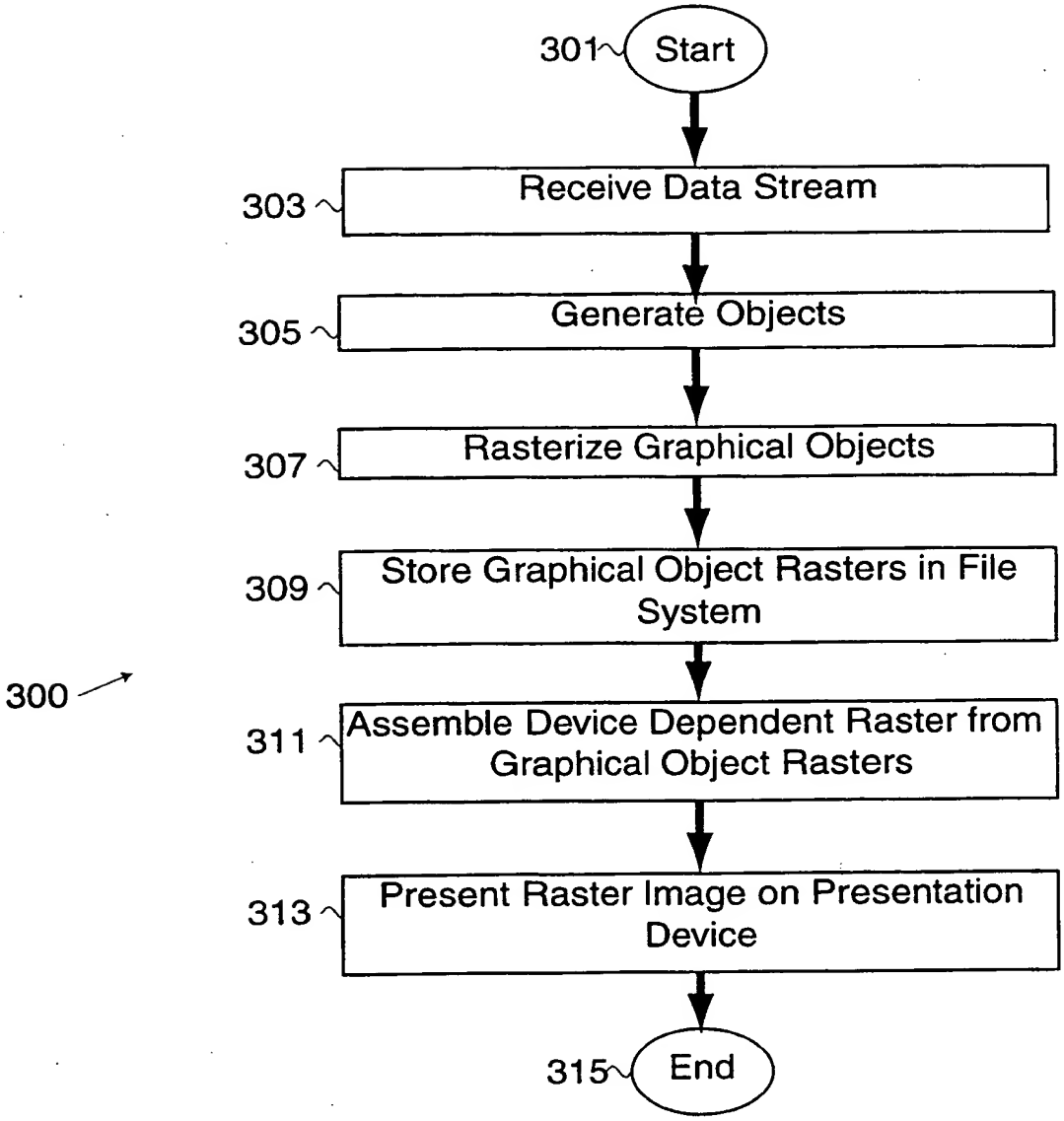

301
Start
303
Receive Data Stream
305
Generate Objects
307
Rasterize Graphical Objects
309
Store Graphical Object Rasters in File System
300
311
Assemble Device Dependent Raster from Graphical Object Rasters
313
Present Raster Image on Presentation Device
315
End

Fig. 3

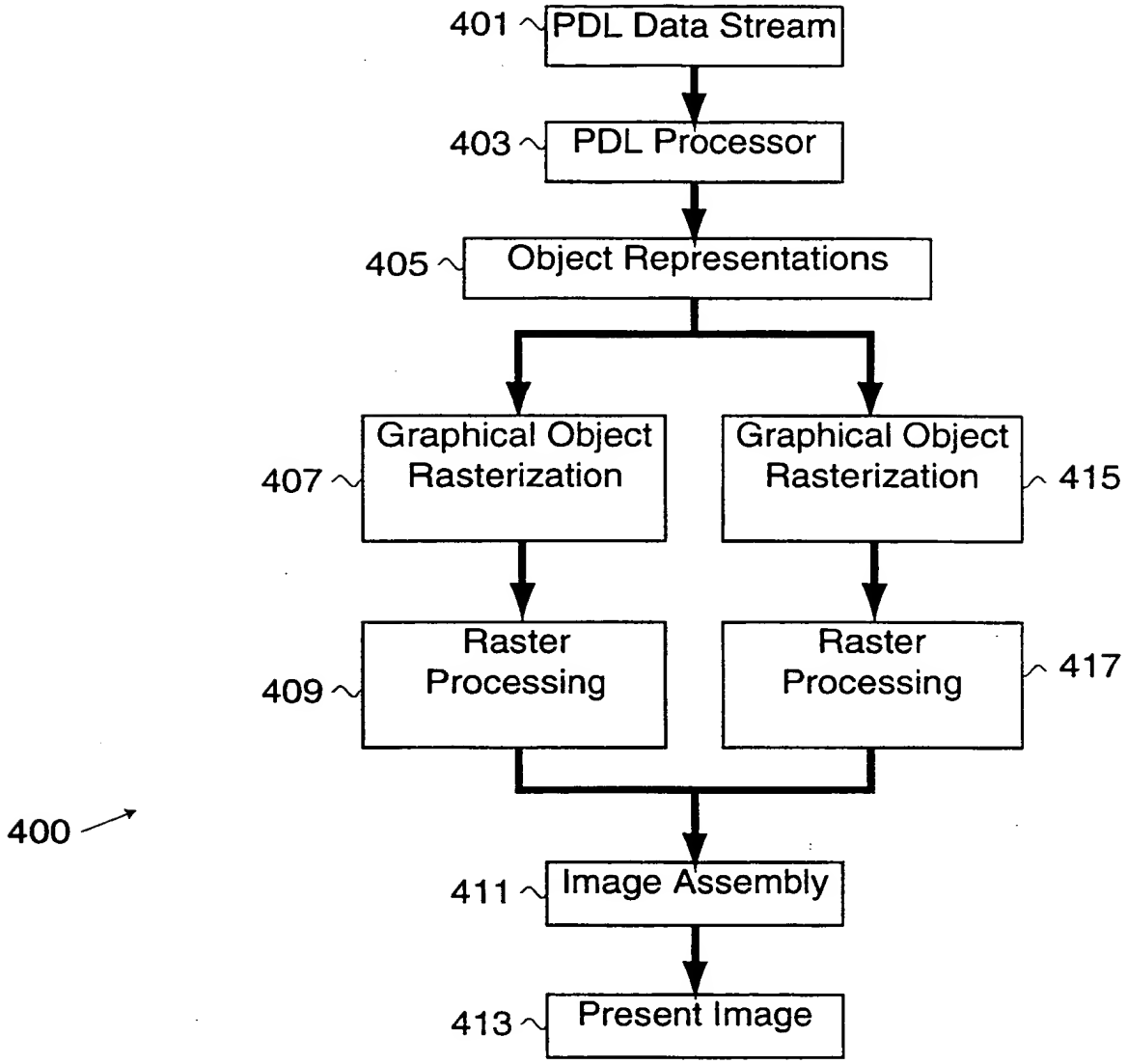

401
PDL Data Stream
403
PDL Processor
405
Object Representations
Graphical Object Rasterization
407
Graphical Object Rasterization
415
Raster Processing
409
Raster Processing
417
400
411
Image Assembly
413
Present Image

Fig. 4A

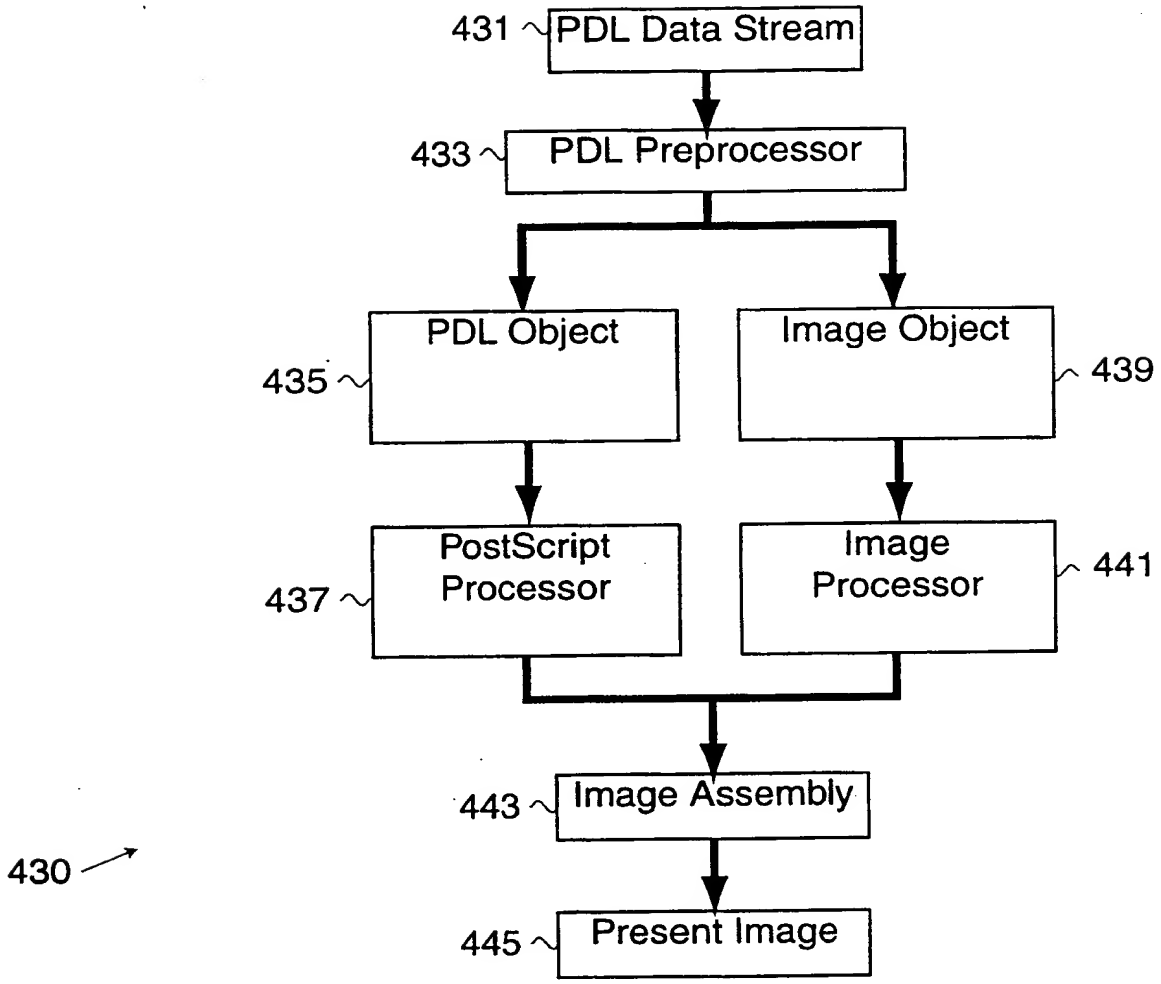

431
PDL Data Stream
433
PDL Preprocessor
PDL Object
435
Image Object
439
PostScript Processor
437
Image Processor
441
443
Image Assembly
445
Present Image
430

Fig. 4B

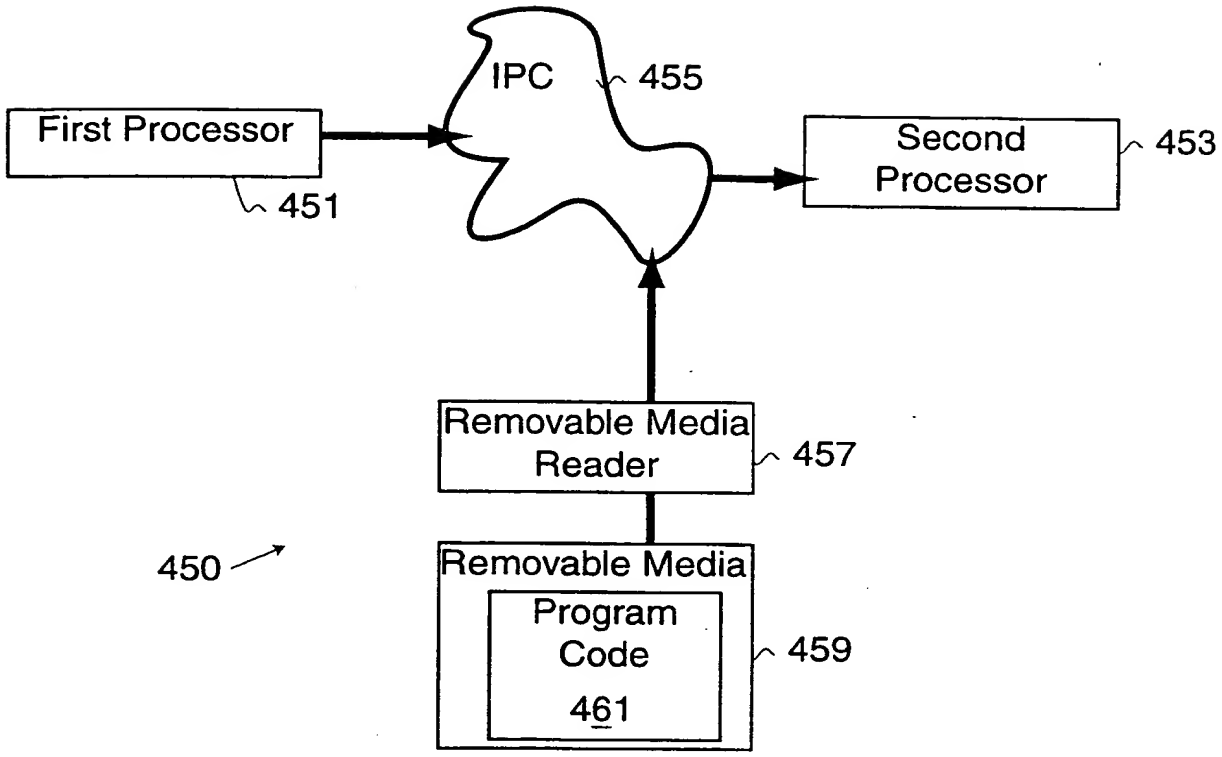

IPC
455
First Processor
451
Second Processor
453
Removable Media Reader
457
450
Removable Media
Program Code
461
459

Fig. 4C

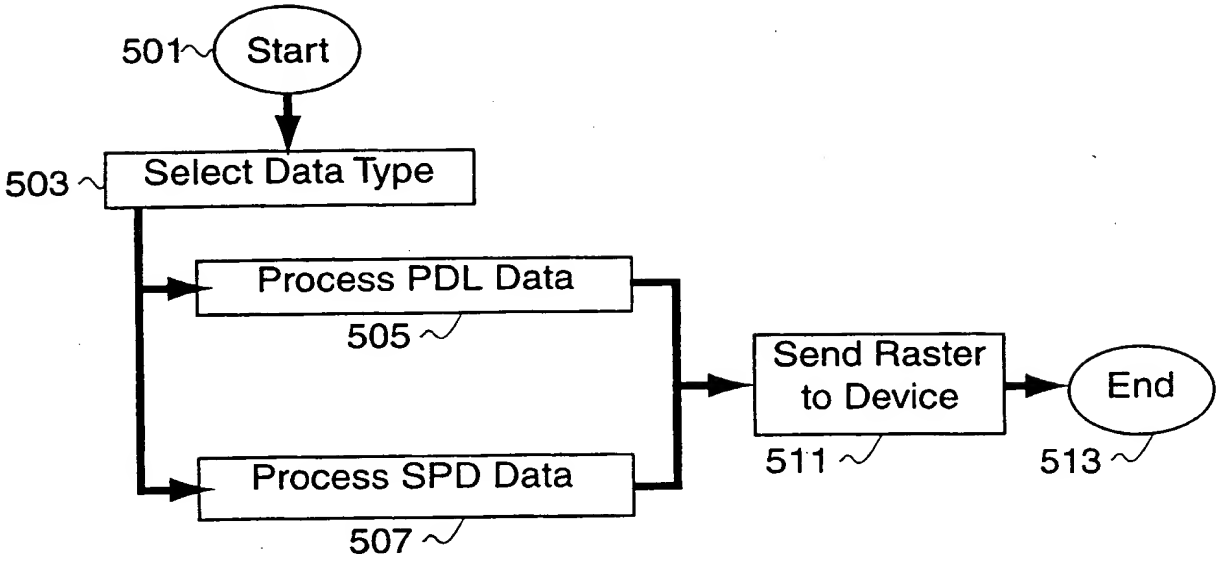
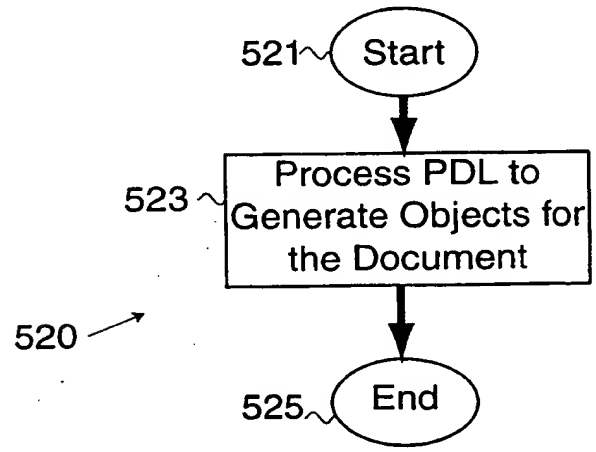

501
Start
503
Select Data Type
Process PDL Data
505
Process SPD Data
507
Send Raster to Device
511
End
513
500

Fig. 5A

Fig. 5B


521
Start
523
Process PDL to Generate Objects for the Document
520
End
525

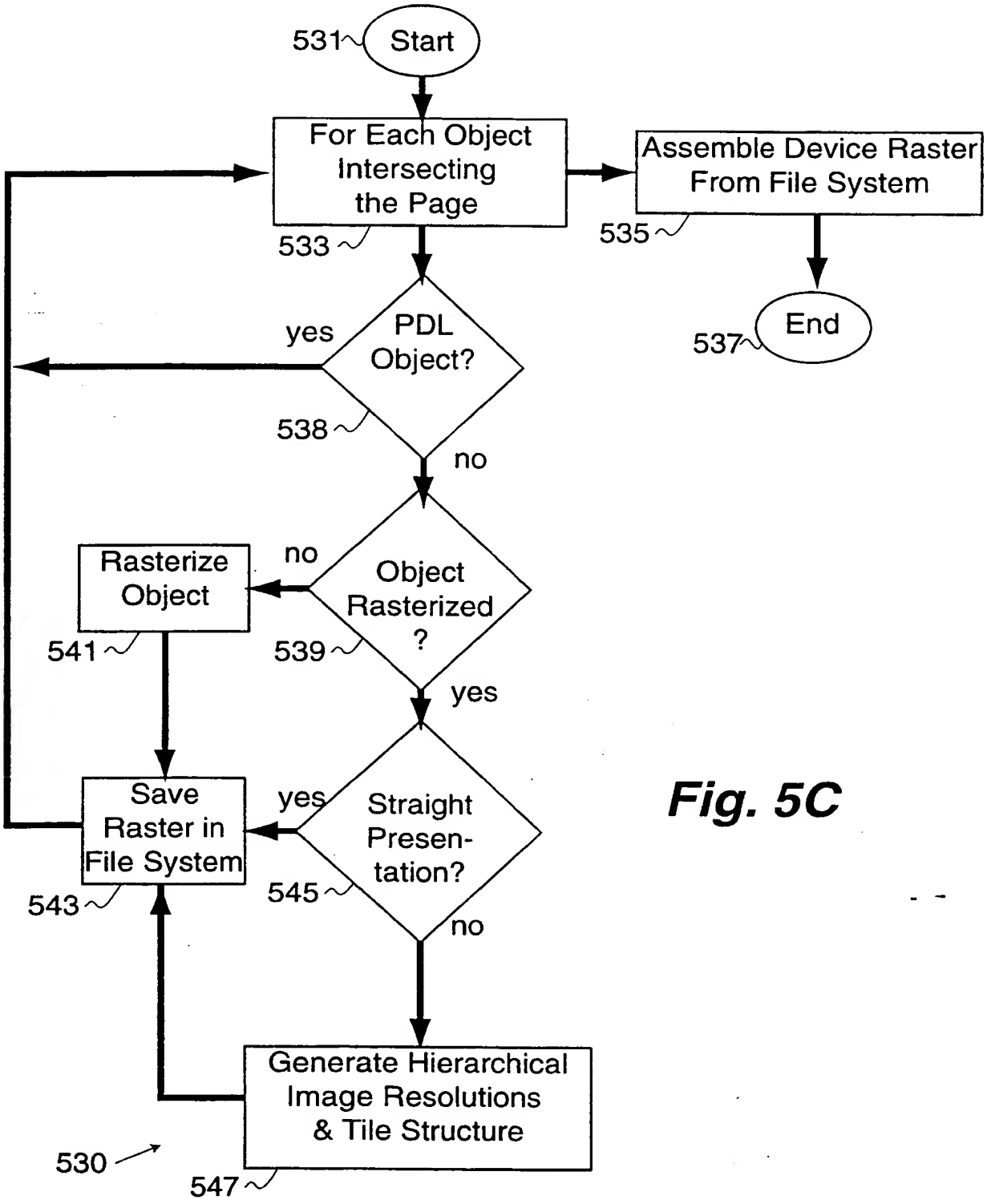

531
Start
For Each Object Intersecting the Page
533
Assemble Device Raster From File System
535
End
537
yes
PDL Object?
538
no
Object Rasterized ?
539
no
Rasterize Object
541
yes
Straight Presen-tation?
545
yes
Save Raster in File System
543
no
Generate Hierarchical Image Resolutions & Tile Structure
547
530

Fig. 5C

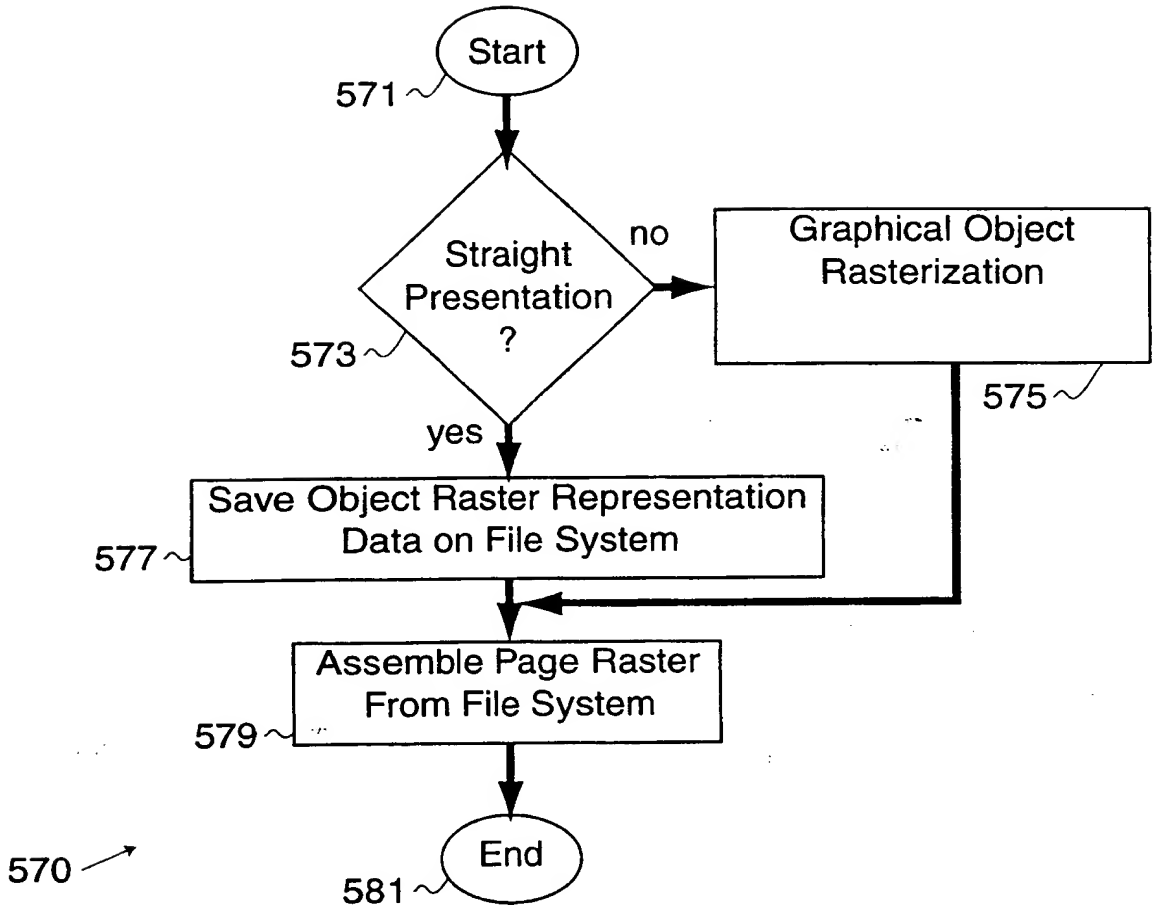

Start
571
Straight Presentation ?
573
no
Graphical Object Rasterization
575
yes
Save Object Raster Representation Data on File System
577
Assemble Page Raster From File System
579
End
581
570

Fig. 5D

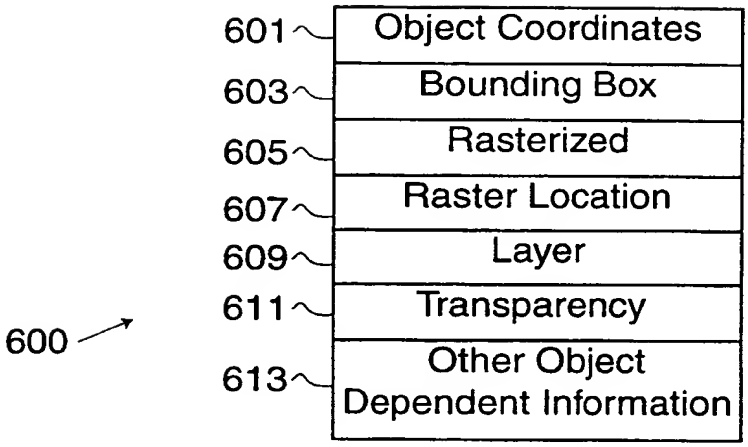

600
601
Object Coordinates
603
Bounding Box
605
Rasterized
607
Raster Location
609
Layer
611
Transparency
613
Other Object Dependent Information

Fig. 6

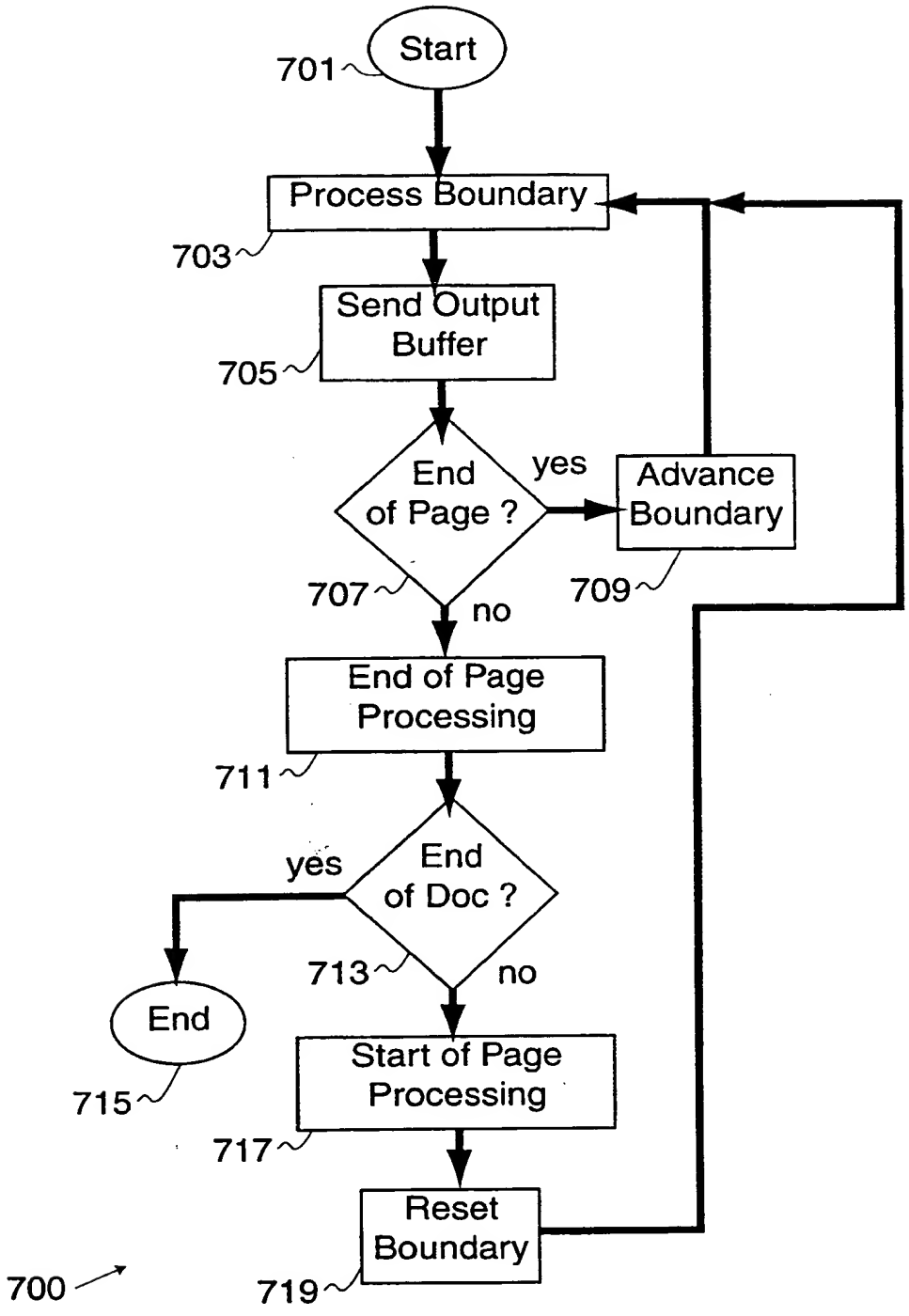

Start
701
Process Boundary
703
Send Output Buffer
705
End of Page ?
yes
Advance Boundary
707
709
no
End of Page Processing
711
End of Doc ?
yes
713
no
End
715
Start of Page Processing
717
Reset Boundary
719
700

Fig. 7A

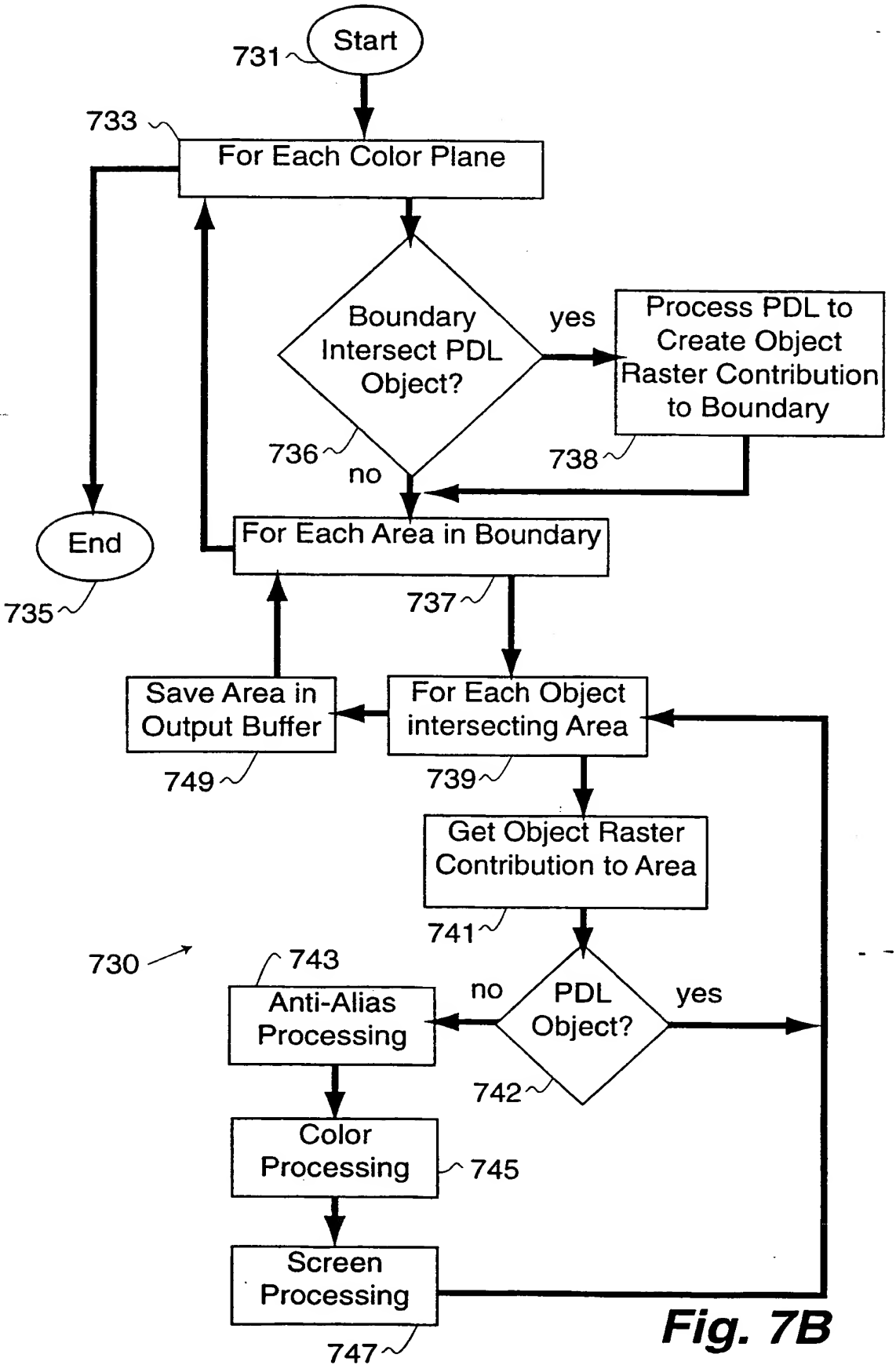

Start
731
733
For Each Color Plane
Boundary Intersect PDL Object?
yes
Process PDL to Create Object Raster Contribution to Boundary
736
no
738
End
735
For Each Area in Boundary
737
Save Area in Output Buffer
749
For Each Object intersecting Area
739
Get Object Raster Contribution to Area
741
730
743
Anti-Alias Processing
no
PDL Object?
yes
742
Color Processing
745
Screen Processing
747

Fig. 7B

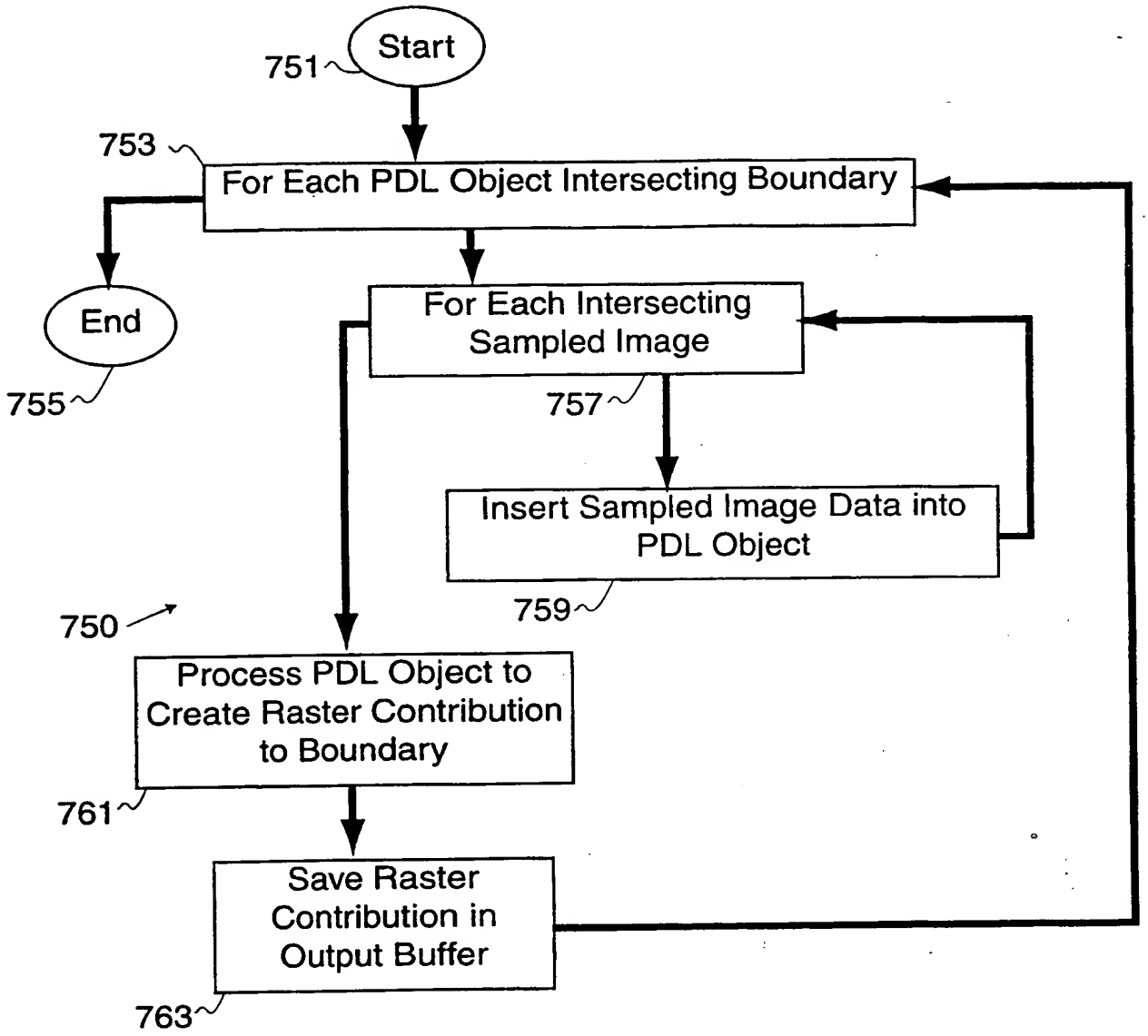

Start
751
753
For Each PDL Object Intersecting Boundary
End
755
For Each Intersecting Sampled Image
757
Insert Sampled Image Data into PDL Object
759
750
Process PDL Object to Create Raster Contribution to Boundary
761
Save Raster Contribution in Output Buffer
763

Fig. 7C

INTERNATIONAL SEARCH REPORT

In. ational Application No
PCT/US 98/21587

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06K15/00 G06F3/12 G06T11/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06K G06F G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 647 921 A (XEROX CORP) 12 April 1995 see page 4, line 11 - page 9, line 17; claims	1-5,12, 14,16, 19-23, 30,32, 34,37, 38, 43-45,48
A	EP 0 432 896 A (IBM) 19 June 1991 see page 5, line 23 - line 42 see page 6, line 17 - page 7, line 20	1-3,5-9, 14-16, 19-27, 32-34, 37-40, 45,48

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
1 March 1999	08/03/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Gélébart, Y

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 98/21587

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 674 277 A (ALDUS CORP) 27 September 1995 ---	
A	US 5 528 704 A (COWARD ROBERT E ET AL) 18 June 1996 see column 4, line 60 - column 5, line 51 ---	1,16,19, 34,37, 45,48
A	EP 0 733 965 A (AGFA GEVAERT NV) 25 September 1996 ---	
A	PATENT ABSTRACTS OF JAPAN vol. 097, no. 010, 31 October 1997 & JP 09 167222 A (FUJI XEROX CO LTD), 24 June 1997 see abstract ---	
A	PATENT ABSTRACTS OF JAPAN vol. 018, no. 574 (P-1821), 2 November 1994 & JP 06 208442 A (CANON INC), 26 July 1994 see abstract ---	
A	PATENT ABSTRACTS OF JAPAN vol. 017, no. 634 (P-1649), 24 November 1993 & JP 05 205013 A (CANON INC), 13 August 1993 see abstract -----	

2

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int tional Application No
PCT/US 98/21587

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 647921 A	12-04-1995	US 5485568 A	16-01-1996
		CA 2131439 A	09-04-1995
		JP 7282284 A	27-10-1995
EP 0432896 A	19-06-1991	CA 2028701 A	16-05-1991
		DE 69032082 D	09-04-1998
		DE 69032082 T	29-10-1998
		JP 2093272 C	18-09-1996
		JP 3161882 A	11-07-1991
		JP 8010465 B	31-01-1996
		US 5157765 A	20-10-1992
EP 0674277 A	27-09-1995	US 5666543 A	09-09-1997
		AU 1500395 A	05-10-1995
		CA 2145216 A	24-09-1995
		JP 8083337 A	26-03-1996
US 5528704 A	18-06-1996	NONE	
EP 0733965 A	25-09-1996	JP 8297560 A	12-11-1996
		US 5652711 A	29-07-1997